

# Delta Programming Task Documentation

## Requirements:

- Allow the user to enter a station (destination or origin) to search flights. Display the results in a table.
- Provide an auto-suggest feature for station.
- Provide two RESTful endpoints supporting the functionality listed in steps 1 and 2.

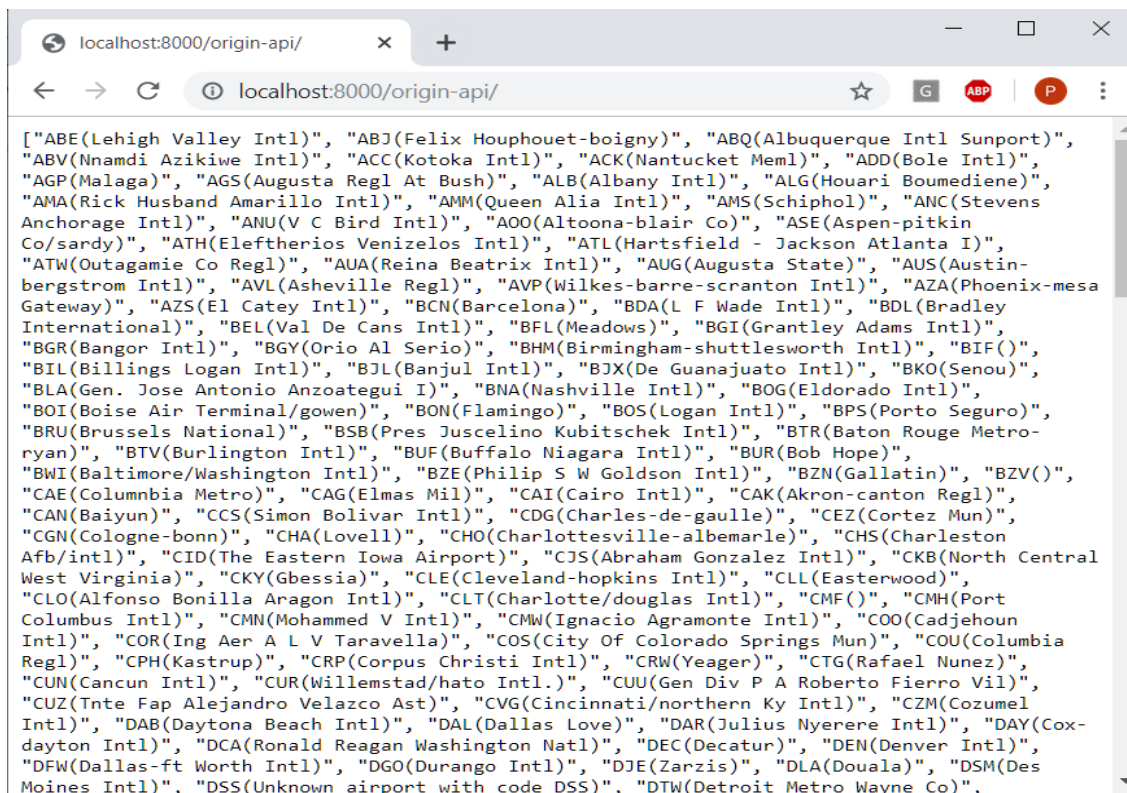
## Implementation

### REST API

Following REST APIs have been implemented to support this implementation:

#### *Origin-api*

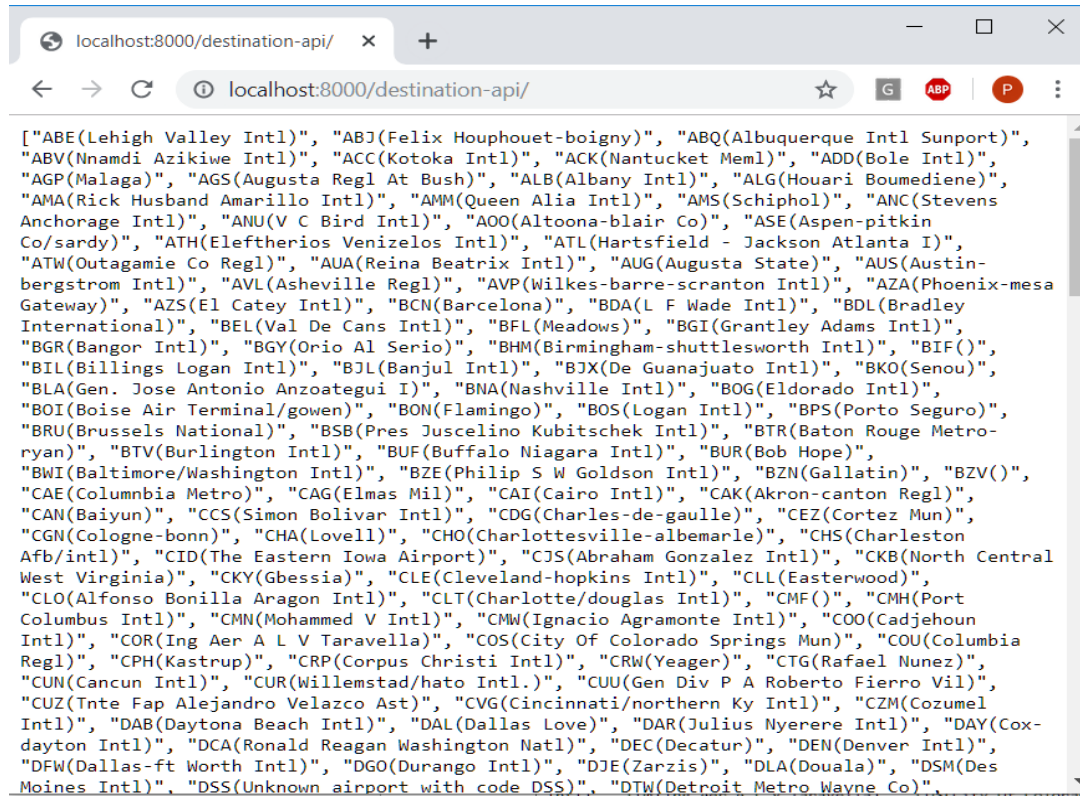
JSON response containing a unique list of all the Origins in the file. Consumed by the web application for the Auto-suggest feature



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/origin-api/'. The main content area shows a long JSON array of airport codes and names, such as '["ABE(Lehigh Valley Intl)", "ABJ(Felix Houphouet-boigny)", "ABQ(Albuquerque Intl Sunport)", ...]'. The list includes various international and domestic airports, ending with 'DTW(Detroit Metro Wayne Co)".

### Destination-api

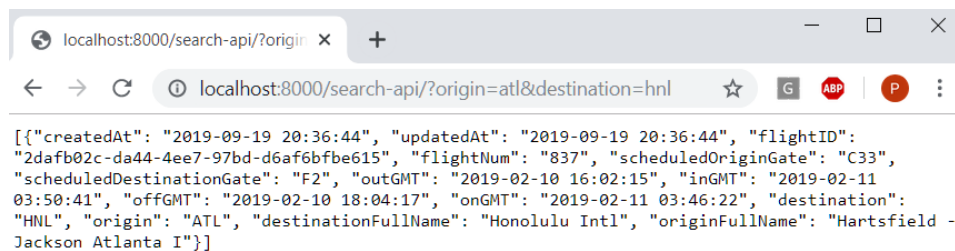
JSON response containing a unique list of all the destinations in the file. Consumed by the web application for the Auto-suggest feature



A screenshot of a web browser window displaying the JSON response of the `destination-api` endpoint. The browser's address bar shows `localhost:8000/destination-api/`. The response is a large array of airport codes and names, such as `"ABE(Lehigh Valley Intl)"`, `"ABJ(Felix Houphouet-boigny)"`, `"ABQ(Albuquerque Intl Sunport)"`, and `"DTW(Detroit Metro Wayne Co)"`. The list includes various international and domestic airports.

### Search-api

JSON response containing the flights returned from the search based on the origin and the destination that was passed

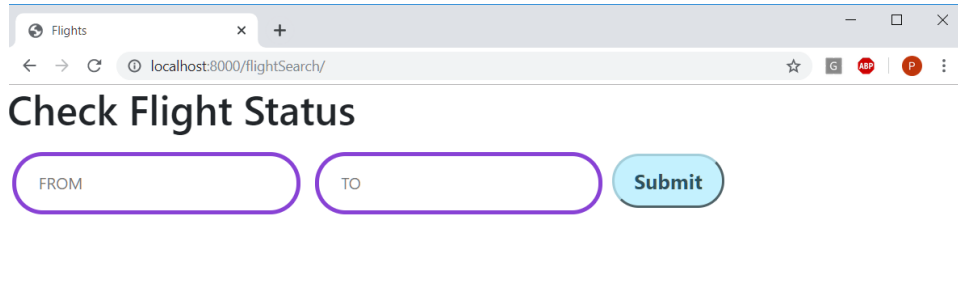


A screenshot of a web browser window displaying the JSON response of the `search-api` endpoint. The browser's address bar shows `localhost:8000/search-api/?origin=atl&destination=hnl`. The response is a JSON object containing flight details: `"createdAt": "2019-09-19 20:36:44"`, `"updatedAt": "2019-09-19 20:36:44"`, `"flightID": "2dafb02c-da44-4ee7-97bd-d6af6bfb615"`, `"flightNum": "837"`, `"scheduledOriginGate": "C33"`, `"scheduledDestinationGate": "F2"`, `"outGMT": "2019-02-10 16:02:15"`, `"inGMT": "2019-02-11 03:50:41"`, `"offGMT": "2019-02-10 18:04:17"`, `"onGMT": "2019-02-11 03:46:22"`, `"destination": "HNL"`, `"origin": "ATL"`, `"destinationFullName": "Honolulu Intl"`, and `"originFullName": "Hartsfield - Jackson Atlanta I"`.

## Web Application

Django based implementation to get a list of flights matching the search criteria. Search can be made using the origin or the destination or both.

localhost:8000/flightSearch/



The screenshot shows a web browser window with the title 'Flights'. The address bar displays 'localhost:8000/flightSearch/'. The main heading is 'Check Flight Status'. Below the heading, there are two input fields: the first is labeled 'FROM' and the second is labeled 'TO'. To the right of these fields is a blue 'Submit' button. The browser's address bar also shows navigation icons (back, forward, refresh) and a star icon for bookmarks. The browser's tab bar shows a single tab titled 'Flights'.

## Auto Suggest

Auto suggest feature automatically suggests the available options for origin and destination as the user types on these fields. Suggestion is based on the values on both the airport code and the full name. Any substring matching the characters typed by the user is available in the suggestion. For example, typing 'at' in the origin field would suggest any of the origins that contain the string 'at' in the origin airport code or the full name.

# Check Flight Status

at ▼

TO

AGS(Augusta Regl At Bush)

ATH(Eleftherios Venizelos Intl)

ATL(Hartsfield - Jackson Atlanta I)

ATW(Outagamie Co Regl)

AUA(Reina Beatrix Intl)

AUG(Augusta State)

AZA(Phoenix-mesa Gateway)

AZS(El Catey Intl)

BDL(Bradley International)

BJX(De Guanajuato Intl)

# Check Flight Status

ATL(Hartsfield - Jackson Atlant

orl ▼

JDO(Orlando Bezerra De Menezes)

MCO(Orlando Intl)

MSY(Armstrong New Orleans Intl)

SFB(Orlando Sanford)

Submit

Search results

**Check Flight Status**

ATL MCO

Flight#	Origin	Destination	Origin Full Name	Destination Full Name	Scheduled Origin Gate	Scheduled Destination Gate	Out GMT	In GMT	Off GMT	On GMT
1158	ATL	MCO	Hartsfield - Jackson Atlanta I	Orlando Intl	A06	76	2019-02-09 21:54:15	2019-02-09 23:42:58	2019-02-09 22:36:52	2019-02-09 23:37:38
1153	ATL	MCO	Hartsfield - Jackson Atlanta I	Orlando Intl	A26	74	2019-02-09 23:00:15	2019-02-10 01:00:53	2019-02-09 23:50:52	2019-02-10 00:56:08
1155	ATL	MCO	Hartsfield - Jackson Atlanta I	Orlando Intl	A18	75	2019-02-10 00:05:39	2019-02-10 02:06:47	2019-02-10 00:58:52	2019-02-10 02:02:32

## Design consideration

Following design decisions were made with this implementation:

- Python Django based implementation – Django provides a rich framework for developing web applications based on the model-template-view architecture
- Security – Application did not implement any security, since the data is not sensitive. Data is sent to the server using 'GET' requests as well.
- Bootstrap CSS framework has been leveraged for the front end
- JQuery/Javascript is used for the auto suggest, form validation and pagination
- Both APIs and the web application are implemented under the same project for simplicity. Ideally, we would want the api to be implemented on its own

## Known issues:

Currently the auto-suggest feature seems slow to pull the suggestions. Will look at ways to optimize this implementation

## Source files:

Source files are laid out based on the Django framework

Models.py - Contains the model class used for this implementation

Views.py - Contains the implementation for all the views (for the api and the web application)

Urls.py - Contains the routing information for the api calls and the web application

Templates – All the html templates used in this implementation are in this folder

Static files – CSS and javascript files are in this folder